#include<stdio.h>

#include"add.h"

#include<stdlib.h>

#include"sub.h"

int sub(int a, int b) {

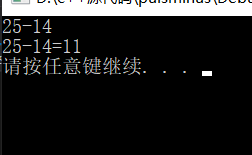
printf("%d-%d=%d\n", a, b, a - b);

}

int add(int a, int b) {

printf("%d+%d=%d\n", a, b, a + b);

}

int main(void) {

char str[20]="\0";

scanf\_s("%s",str,20);

int a, b,i=0,temp,num1=0,num2=0;

while (str[i] <= '9'&&str[i] >= '0') {

if (str[i] <= '9'&&str[i] >= '0') {

num1 = num1\*10+str[i] - '0';

}

i++;

}

temp = i;

i++;

while (i<strlen(str)) {

if (str[i] <= '9'&&str[i] >= '0') {

num2 = num2 \* 10 + str[i] - '0';

}

i++;

}

if (str[temp] == '+') {

add(num1, num2);

}

else if (str[temp] == '-') {

sub(num1, num2);

}

system("pause");

return 0;

}

|  |  |
| --- | --- |
| #include<stdio.h>  #include<stdlib.h>  #include"student.h"  int main() {  Node \* NodeHead = NULL;  int nSelection;  int num;  char rubbish[1000] = { 0 };  nSelection = -1;  printf("========学生成绩管理系统==============");  do {  printf("\n1.Append record\n");  printf("2.List record\n");  printf("3.Delete record\n");  printf("4.Modify record\n");  printf("5.Search record\n");  printf("6.Sort in descending order by sum\n");  printf("7.Sort in ascending order by sum\n");  printf("8.Sort in descending order by num\n");  printf("9.Sort in ascending order by num\n");  printf("请选择菜单项编号（0~3）：");  nSelection = -1;  num = scanf\_s("%d", &nSelection);  switch (nSelection) {  case 1: {  addNode(&NodeHead);  break;  }  case 2: {  displayNode(NodeHead);  break;  }  case 3: {  Delete(&NodeHead);  break;  } | case 4: {  Modify(NodeHead);  break;  }  case 5: {  search(NodeHead);  break;  }  case 6: {  descsortsum(NodeHead);  break;  }  case 7: {  ascsortsum(NodeHead);  break;  }  case 8: {  descsortnum(NodeHead);  break;  }  case 9: {  ascsortsum(NodeHead);  break;  }  case 0:  {  printf("谢谢您使用本系统！");  exit(0);  }  default:  {  printf("输入的菜单编号错误！\n");  break;  }  }  } while (nSelection != 0);  return 0;  } |

|  |  |
| --- | --- |
| void addNode(Node\*\* pNodeHead) {  char name[10];//添加结点  int num;  int men=2;  float score;  Node\* stuNode = malloc(sizeof(Node));//构成链表  float sum=0,average;//总分,平均分  int flag = 0;  Node\*ppNodeHead = \*pNodeHead;  Node\*ppNodeTail = \*pNodeHead;  if (Inputname(name) && Inputnum(&num) && Inputscore(&score)) {  stuNode->score = score;  strcpy(stuNode->name, name);  stuNode->num = num;  if (ppNodeHead == NULL) {  //第一张卡  ppNodeHead = stuNode;  ppNodeTail = stuNode;  \*pNodeHead = stuNode;  sum += stuNode->score;  }  else {  ppNodeTail = ppNodeHead;  while (ppNodeTail->next != NULL) {//找到链表最后一个结点  sum += ppNodeTail->score;  men++;  ppNodeTail = ppNodeTail->next;  }  sum += ppNodeTail->score;  sum += stuNode->score;  ppNodeTail->next = stuNode;  ppNodeTail = stuNode;//尾  void displayNode(Node\* NodeHead) {  if (NodeHead == NULL) {  printf("\n没有信息！\n");  return;  }  printf("姓名\t学号\t分数\n");  Node\* Cur = NodeHead;//当前结点进行遍历  while (Cur != NULL) {  printf("%s\t%d\t%f\n", Cur->name, Cur->num, Cur->score);  Cur = Cur->next;  }  }  void search(Node\* NodeHead) {  char name[10];  Inputname(name);  Node\* pNode = NodeHead;  printf("姓名\t学号\t分数\n");  int find = false;  while (pNode != NULL) {  if (strstr(pNode->name, name)!=NULL) {  find = true;  printf("%s\t%d\t%f\n", pNode->name, pNode->num, pNode->score);  }  pNode = pNode->next;  }  if (!find) {  printf("NULL\tNULL\tNULL\n");  printf("未找到与关键字相似的学生！");  }  } | 指针指导链表表尾  }  ppNodeTail->next = NULL;  flag = 1;  if (flag) {  printf("\n\n添加新卡成功！\n");  printf("姓名\t学号\t分数\n");  printf("%s\t%d\t%f", name, num, score);  printf("\n总分\t平均分\n");  printf("%f\t%f\n", sum,sum/men);  }  else {  printf("\n\n系统错误！\n");  }  }else {  printf("\n输入的【姓名，学号，分数】等信息格式不符合要求，添加卡失败！\n");  }  }  void Modify(Node\* NodeHead) {  int num;//修改结点  Inputnum(&num);  Node\* pNode = NodeHead;  printf("姓名\t学号\t分数\n");  int find = false;  while (pNode != NULL) {  if (pNode->num == num) {  find = true;  modify(pNode);  printf("%s\t%d\t%f\n", pNode->name, pNode->num, pNode->score);  break;  }  pNode = pNode->next;  }  return;}  void Delete(Node\*\* NodeHead) {  int num;  Inputnum(&num);  Node\* pNode = \*NodeHead;  printf("姓名\t学号\t分数\n");  int find = false;  Node\* temp =pNode;  int i = 0;  while (pNode != NULL) {  if (pNode->num == num) {  find = true;  if (i == 0) {  \*NodeHead = pNode->next;  free(pNode);  break;  }  else {  temp->next = pNode->next;  free(pNode);  // printf("%s\t%d\t%f\n", pNode->name, pNode->num, pNode->score);  break;  }  }  temp = pNode;  i++;  pNode = pNode->next;  }  } |

|  |  |
| --- | --- |
| //按总分降序排列 其他类似不再给出  void descsortsum(Node\*NodeHead) {  Node \*cur, \*tail;  cur = NodeHead;  tail = NULL;  float ftemp;  int itemp;  char stemp[10];  if (cur == NULL || cur->next == NULL) {//如果只有一个结点或没有结点就不需要排序了  return;  }  while (cur->next != tail) {  while (cur->next != tail) {//判断当前指针是否到达最后一个  if (cur->score <cur->next->score) {  ftemp = cur->score;  cur->score = cur->next->score;  cur->next->score = ftemp;  itemp = cur->num;  cur->num = cur->next->num;  cur->next->num = itemp;  strcpy(stemp, cur->name);  strcpy(cur->name, cur->next->name); strcpy(cur->next->name, stemp);  }  cur = cur->next;  }  tail = cur;//保存当前的结点 相当于头结点  cur =NodeHead ;//从当前的下一个开始  }} | 按降序排序后 |

通讯录程序与学生管理程序几乎相同，这里贴出上一程序用到的一些函数

|  |  |
| --- | --- |
| typedef struct node {  char name[10];  char phone[12];  struct node \*next;  }Node;  int Inputname(char \*name) {  while (true) {  printf("请输入姓名（长度为1~5）:");  scanf\_s("%s", name, 10);  if (strlen(name) >= 10) {  printf("卡号长度已超过5位！");  char ch = 'M';  while (ch != 'n'&&ch != 'y') {  printf("重新输入吗？（y/n);");  scanf\_s("%c", &ch);  }  if (ch == 'n') {  return false;  }  }  else {  return true;  break;  }  }  } | int Inputnum(char \*num) {  while (true) {  printf("请输入电话:");  scanf\_s("%s", num,12);  if (strlen(num)>11) {  printf("卡号长度已超过11位！");  char ch = 'M';  while (ch != 'n'&&ch != 'y') {  printf("重新输入吗？（y/n);");  scanf\_s("%c", &ch);  }  if (ch == 'n') {  return false;  }  }  else {  return true;  break;  }  }  } |

|  |  |
| --- | --- |
| #include<stdio.h>  int main(){  int number,num[3];  scanf("%d",&number);  printf("%s%d%s","整数",number,"的");  for(int i=0;i<3;i++){  num[i]=number%10;  number/=10;  }  printf("%s%d","个位数字是",num[0]);  printf("%s%d",",十位数字是",num[1]);  printf("%s%d","，百位数字是",num[2]);  } | #include<stdio.h>  int main(){  int days;  scanf("%d",&days);  int n=days%5;  if(n<=3){  printf("%s%d","Fishing in day ",days);  }  else{  printf("%s%d","Drying in day ",days);  }  } |

|  |  |
| --- | --- |
| #include<stdio.h>  int main(){  double money,tax;  scanf("%lf",&money);  if(money<=1600.00){  tax=0;  }  else if(money<2500.00){  tax=0.05\*(money-1600);  }  else if(money<3500.00){  tax=0.1\*(money-1600);  }  else if(money<4500.00){  tax=0.15\*(money-1600);  }  else {  tax=0.2\*(money-1600);  }  printf("%.2f",tax);  } |  |

|  |  |
| --- | --- |
| #include<stdio.h>  int main(){  int a,n,total;  scanf("%d%d",&a,&n);  total=a;  int temp=a;  for(int i=1;i<n;i++){  a\*=10;  a+=temp;  total+=a;  }  printf("%d",total);  } | #include<stdio.h>  #include<string.h>  #include<stdlib.h>  int main(){  char s[100];  int i=0,num,english,space,other;  english=num=space=other=0;  do{  scanf("%c",&s[i]);  if((s[i]>='a'&&s[i]<='z')||(s[i]>='A'&&s[i]<='Z')){  english++;}  else if(s[i]==' '){  space++;}  else if((s[i]-'0')<=9&&(s[i]-'0')>=0){  num++; }  else if(s[i]!='#'){  other++;}  i++;  }while(s[i-1]!='#');  printf("%s%d\n","英文字母个数为",english);  printf("%s%d\n","空格个数为",space);  printf("%s%d\n","数字字符个数为",num);  printf("%s%d\n","其他字符个数为",other);  } |

|  |  |
| --- | --- |
| #include<stdio.h>  int main(){  for(int i=2;i<=1000;i++){  int sum=0;  for(int j=1;j<i;j++){  if(i%j==0){  sum+=j; printf("%d ",j);} }  if(sum==i){  printf("%d \n",i);  }}} |  |

|  |  |
| --- | --- |
| #include<stdio.h>  int main(){  double money;  scanf("%lf",&money);  while(money>1.0||money<0.08){  printf("%s","请重新输入："); scanf("%lf\n",&money);  }  int coin[5000],n=0,kind=0;  for(int i=0;i<=(int)money/0.01;i++){  for(int j=0;j<=(int)money/0.02;j++){  for(int m=0;m<=(int)money/0.05;m++){  if(0.01\*i+0.02\*j+0.05\*m==money){  coin[n]=i;  n++;  coin[n]=j;  n++;  coin[n]=m;  n++;  kind++;  }  }  }  } | int total=kind;  while(kind){  kind--;  n--;  int total=0;  printf("%s%d\n","五分硬币数量：",coin[n]);  total+=coin[n];  n--;  printf("%s%d\n","二分硬币数量：",coin[n]);  total+=coin[n];  n--;  printf("%s%d\n","一分硬币数量：",coin[n]);  total+=coin[n];  printf("%s%d\n\n","硬币总数量：",total);  }  printf("%s%d%s\n","共有",total,"种换法");  } |

|  |  |
| --- | --- |
| #include<stdio.h>  int countDigit(int number,int digit){  int num=0;  while(number){  if(number%10==digit){  num++;  }  number/=10;  }  return num;  }  int main(){  int num;  scanf("%d",&num);  printf("Number of digit 2:%d",countDigit(num,2));  } | #include<stdio.h>  int IsSquare(int n){  int i;  for(i=1;i\*i<=n;i++){  if(i\*i==n){  return 1;  }  }  return 0;  }  int main(){  int x;  printf("Input x(int x>0)...\nx=");  if(scanf("%d",&x)!=1 || x<1){  printf("Input error, exit...\n");  return 0;  }  printf(IsSquare(x) ? "YES\n" : "NO\n");  return 0;  } |

|  |  |
| --- | --- |
| #include<stdio.h>  int main(){  int n;scanf("%d",&n);  int a[10];  for(int i=0;i<n;i++){  scanf("%d",&a[i]); }  int temp;  for(int i=0;i<n/2;i++){  temp=a[i];a[i]=a[n-1-i];a[n-1-i]=temp;  }  for(int i=0;i<n;i++){  printf("%d ",a[i]);  }} |  |
| #include<stdio.h>  int main(){  int n;  scanf("%d",&n);  int a[10];  for(int i=0;i<n;i++){  scanf("%d",&a[i]);  }  int b[9];  for(int i=0;i<n;i++){  if(i!=n-1){  b[i]=a[i+1]-a[i];  }  }  for(int i=0;i<n-1;++i){  if((i+1)%3!=0){  printf("%d\t",b[i]);  }  else{  printf("%d\n",b[i]);  }  }  } | #include<stdio.h>  int main(){  int a[10][10]={{5,53,74,58},  {79,85,25,82},  {92,4,16,87}};  int b[10];  for(int i=0;i<10;i++){  int max=a[i][0];  for(int j=0;j<10;j++){  if(a[i][j]>max){  max=a[i][j];  }  }  b[i]=max;  printf("%d\t",max);  }    } |

|  |  |
| --- | --- |
| #include<stdio.h>  #include<assert.h>  int main(){  char des[20]="guoyawen";  char src[20]="wutao";  char \*str=my\_strcat(des,src);  printf("%s\n",str);  return 0;  } | char\* my\_strcat(char \*des,const char \*src){  assert(des!=NULL&&src!=NULL);  char \*temp=des;  while(\*temp!='\0'){  temp++;}  while(\*src){  \*temp=\*src;temp++;src++;  }  \*temp='\0';  return des;  } |